

## Task1

### MAE(Mean Absolute Error):

$$MAE = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{n}$$

其中 $y_i$ 代表第 $i$ 个样本的真实值, 其中 $\hat{y}_i$ 代表第 $i$ 个样本的预测值。

分类算法常见的评估指标如下:

- 对于二类分类器/分类算法, 评价指标主要有accuracy, [Precision, Recall, F-score, Pr曲线], ROC-AUC曲线。
- 对于多类分类器/分类算法, 评价指标主要有accuracy, [宏平均和微平均, F-score]。

宏平均和微平均应该就是macro和micro了

平均绝对误差 平均绝对误差 (Mean Absolute Error, MAE): 平均绝对误差, 其能更好地反映预测值与真实值误差的实际情况, 其计算公式如下:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|$$

均方误差 均方误差 (Mean Squared Error, MSE), 均方误差, 其计算公式为:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$R^2$  (R-Square) 的公式为: 残差平方和:

$$SS_{res} = \sum (y_i - \hat{y}_i)^2$$

总平均值:

$$SS_{tot} = \sum (y_i - \bar{y})^2$$

其中 $\bar{y}$ 表示 $y$ 的平均值 得到 $R^2$ 表达式为:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

$R^2$ 用于度量因变量的变异中可由自变量解释部分所占的比例, 取值范围是 0~1,  $R^2$ 越接近1, 表明回归平方和占总平方和的比例越大, 回归线与各观测点越接近, 用 $x$ 的变化来解释 $y$ 值变化的部分就越多, 回归的拟合程度就越好。所以 $R^2$ 也称为拟合优度 (Goodness of Fit) 的统计量。

$y_i$ 表示真实值,  $\hat{y}_i$ 表示预测值,  $\bar{y}$ 表示样本均值。得分越高拟合效果越好。

### 一些指标的应用

### 1.3.3 回归指标评价计算示例

```
1 # coding=utf-8
2 import numpy as np
3 from sklearn import metrics
4
5 # MAPE需要自己实现
6 def mape(y_true, y_pred):
7     return np.mean(np.abs((y_pred - y_true) / y_true))
8
9 y_true = np.array([1.0, 5.0, 4.0, 3.0, 2.0, 5.0, -3.0])
10 y_pred = np.array([1.0, 4.5, 3.8, 3.2, 3.0, 4.8, -2.2])
11
12 # MSE
13 print('MSE:', metrics.mean_squared_error(y_true, y_pred))
14 # RMSE
15 print('RMSE:', np.sqrt(metrics.mean_squared_error(y_true, y_pred)))
16 # MAE
17 print('MAE:', metrics.mean_absolute_error(y_true, y_pred))
18 # MAPE
19 print('MAPE:', mape(y_true, y_pred))
```

MSE: 0.2871428571428571

RMSE: 0.5358571238146014

MAE: 0.4142857142857143

MAPE: 0.1461904761904762

```
1 ## R2-score
2 from sklearn.metrics import r2_score
3 y_true = [3, -0.5, 2, 7]
4 y_pred = [2.5, 0.0, 2, 8]
5 print('R2-score:', r2_score(y_true, y_pred))
```

R2-score: 0.9486081370449679

很感谢赛题理解的分享，感觉很有用，值得记录

作为切入一道赛题的基础，赛题理解是极其重要的，对于赛题的理解甚至会影响后续的特征工程构建以及模型的选择，最主要是会影响后续发展工作的方向，比如挖掘特征的方向或者存在解决问题的方向，对了赛题背后的思想以及赛题业务逻辑的清晰，也很有利于花费更少时间构建更为有效的特征模型，赛题理解要达到的地步是什么呢，把一道赛题转化为一种宏观理解的解决思路。以下将从多方面对于此进行说明：

- 1) 赛题理解究竟是理解什么：理解赛题是不是把一道赛题的背景介绍读一遍就OK了呢？并不是的，理解赛题其实也是从直观上梳理问题，分

析问题是否可行的方法，有多少可行度，赛题做的价值大不大，理清一道赛题要从背后的赛题背景引发的赛题任务理解其中的任务逻辑，可能对于赛题有意义的外在数据有哪些，并对于赛题数据有一个初步了解，知道现在和任务的相关数据有哪些，其中数据之间的关联逻辑是什么样的。对于不同的问题，在处理方式上的差异是很大的。如果用简短的话来说，并且在比赛的角度或者做工程的角度，就是该赛题符合的问题是什么问题，大概要去用哪些指标，哪些指标是否会做到线上线下的一致性，是否有效的利于我们进一步的探索更高线上分数的线下验证方法，在业务上，你是否对很多原始特征有很深刻的了解，并且可以通过EDA来寻求他们直接的关系，最后构造出满意的特征。

- 2) 有了赛题理解后能做什么：在对于赛题有了一定的了解后，分析清楚了问题的类型性质和对于数据理解的这一基础上，是不是赛题理解就做完了呢？并不是的，就像摸清了敌情后，我们至少就要有一些相应的理解分析，比如这题的难点可能在哪里，关键点可能在哪里，哪些地方可以挖掘更好的特征，用什么样得线下验证方式更为稳定，出现了过拟合或者其他问题，估摸可以用什么方法去解决这些问题，哪些数据是可靠的，哪些数据是需要精密的处理的，哪部分数据应该是关键数据（背景的业务逻辑下，比如CTR的题，一个寻常顾客大体会有怎么样的购买行为逻辑规律，或者风电那种题，如果机组比较邻近，相关一些风速，转速特征是否会很近似）。这时是在一个宏观的大体下分析的，有助于摸清整个题的思路脉络，以及后续的分析方向。

- 3) 赛题理解的-评价指标：为什么要把这部分单独拿出来呢，因为这部分会涉及后续模型预测中两个很重要的问题：1. 本地模型的验证方式，很多情况下，线上验证是有一定的时间和次数限制的，所以在比赛中构建一个合理的本地的验证集和验证的评价指标是很关键的步骤，能有效的节省很多时间。2. 不同的指标对于同样的预测结果是具有误差敏感的差异性的，比如AUC, logloss, MAE, RSME, 或者一些特定的评价函数。是会有很大可能会影响后续一些预测的侧重点。

- 4) 赛题背景中可能潜在隐藏的条件：其实赛题中有些说明是很有利益-都可以在后续答辩中以及问题思考中所体现出来的，比如高效性要求，

比如对于数据异常的识别处理，比如工序流程的差异性，比如模型运行的时间，比模型的鲁棒性，有些的意识是可以贯穿问题思考，特征，模型以及后续处理的，也有些会对于特征构建或者选择模型上有很大益处，反过来如果在模型预测效果不好，其实有时也要反过来思考，是不是赛题背景有没有哪方面理解不清晰或者什么其中的问题没考虑到。